

# gRPC Messaging



**Michael VanSickle**

Author

@vansimke



# Overview



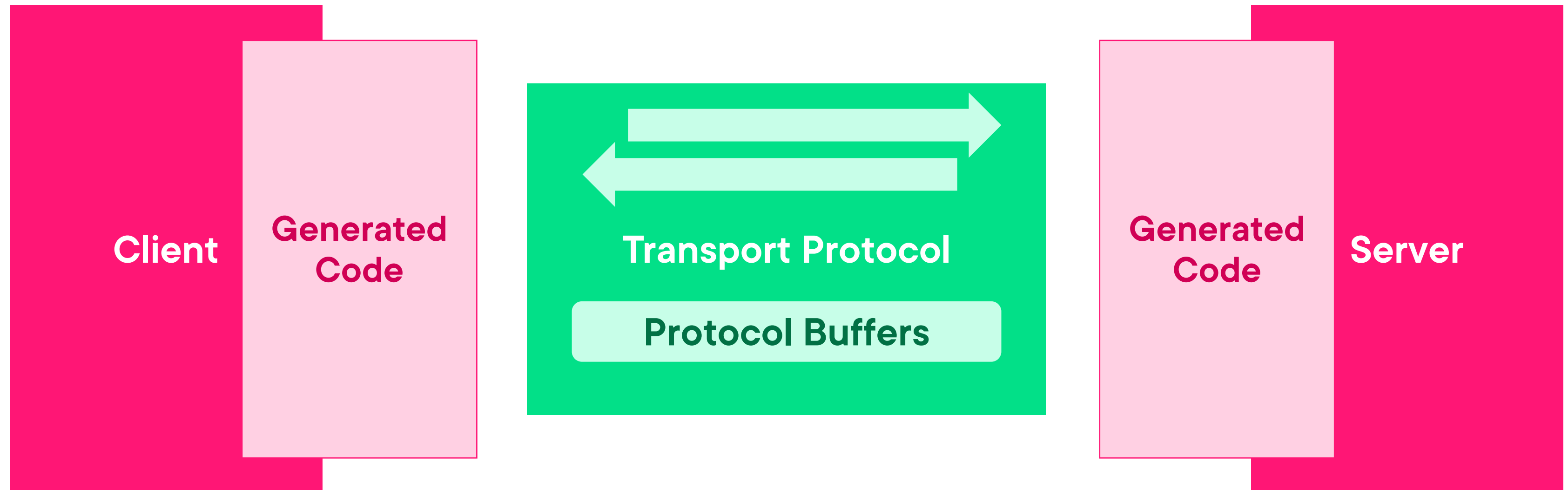
**What is gRPC?**

**Using Protocol Buffers**

**Creating gRPC Clients and Servers**



# gRPC Structure



# Protocol Buffers 101

```
syntax = "proto3";  
package product;  
option go_package = "demo/productpb";  
message Product {  
    int32 id = 1;  
    string name = 2;  
    double usdPerUnit = 3;  
    string unit = 4;  
}
```



# Protocol Buffers 101

```
syntax = "proto3";  
package product;  
option go_package = "demo/productpb";  
message Product {  
    int32 id = 1;  
    string name = 2;  
    double usdPerUnit = 3;  
    string unit = 4;  
}
```



# Protocol Buffers 101

```
syntax = "proto3";  
package product;  
option go_package = "demo/productpb";  
message Product {  
    int32 id = 1;  
    string name = 2;  
    double usdPerUnit = 3;  
    string unit = 4;  
}
```



# Protocol Buffers 101

```
syntax = "proto3";  
package product;  
option go_package = "demo/productpb";  
message Product {  
    int32 id = 1;  
    string name = 2;  
    double usdPerUnit = 3;  
    string unit = 4;  
}
```



# Protocol Buffers 101

```
syntax = "proto3";  
package product;  
option go_package = "demo/productpb";  
message Product {  
    int32 id = 1;  
    string name = 2;  
    double usdPerUnit = 3;  
    string unit = 4;  
}
```





# Protocol Buffers 101

```
syntax = "proto3";  
package product;  
option go_package = "demo/productpb";  
message Product {  
    int32 id = 1;  
    string name = 2;  
    double usdPerUnit = 3;  
    string unit = 4;  
}
```



# Protocol Buffers 101

```
https://protobuf.dev/downloads           //obtain protoc compiler  
  
go install google.golang.org/protobuf/cmd/protoc-gen-go@latest  
                                           // install go package for code generation  
  
protoc -I=. -go_out=. /product.proto    // generate source code
```

<https://protobuf.dev>



# gRPC Message Types

```
syntax = "proto3";
```

```
...
```



# gRPC Message Types

```
syntax = "proto3";
```

```
...
```

```
service Product {
```

```
}
```



# gRPC Message Types

```
syntax = "proto3";  
...  
service Product {  
    rpc  
}
```



# gRPC Message Types

```
syntax = "proto3";  
  
...  
service Product {  
    rpc GetProduct  
}
```



# gRPC Message Types

```
syntax = "proto3";  
  
...  
service Product {  
    rpc GetProduct (Request)  
}
```



# gRPC Message Types

```
syntax = "proto3";  
  
...  
service Product {  
    rpc GetProduct (Request) returns (Response);  
}
```





# gRPC Message Types



**ProductService**

**Client Stub**

**Server Interface**



# Generating gRPC Client and Server Stubs

```
go install google.golang.org/grpc/cmd/protoc-gen-go-grpc@latest
// install go package for code generation

protoc -I=. -go_out=. -go-grpc_out=. /product.proto
// generate source code
```



# Generating gRPC Client and Server Stubs

```
type ProductClient interface {  
    GetProduct(ctx context.Context, in *GetProductRequest, opts ...grpc.CallOption)  
        (*GetProductReply, error)  
}
```



# Generating gRPC Client and Server Stubs

```
type ProductClient interface {  
    GetProduct(ctx context.Context, in *GetProductRequest, opts ...grpc.CallOption)  
        (*GetProductReply, error)  
}
```

```
client := NewProductClient({connection})
```



# Generating gRPC Client and Server Stubs

```
type ProductClient interface {  
    GetProduct(ctx context.Context, in *GetProductRequest, opts ...grpc.CallOption)  
        (*GetProductReply, error)  
}
```

```
client := NewProductClient({gRPC connection})
```

```
type ProductServer interface {  
    GetProduct(context.Context, in *GetProductRequest) (*GetProductReply, error)  
}
```

<https://grpc.io>



# Course Outline



**Creating HTTP Services**

**Serving Static Content**

**JSON Messaging**

**Routing Requests**

**Creating Middleware**

**gRPC Messaging**

